83000.946

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR ANALYZING DATA

INVENTORS:

JEFFREY A. HERMAN
KEVIN T. LOONEY

PREPARED BY:

HECKER & HARRIMAN
2029 Century Park East
Suite 1600
Los Angeles, CA 90067

(310) 286-0377

BACKGROUND OF THE INVENTION

1.   FIELD OF THE INVENTION

5   The present invention relates to a method and apparatus for analyzing data.

2.   BACKGROUND ART

10   Often it is desired to compare one type of data to another type of data. One example is when it is desired to determine whether a person can perform a particular task. In that case, data comprising a list of the skills that the person possesses are compared to data comprising a list of the skills required for the task. Another example is when it is desired to determine the compatibility of a

15   computer program with a particular platform or environment. In that case, data comprising the computing resources required by the computer program is compared to data comprising the resources available on the platform or from the environment.

20   When writing applications for environments, it is important to be able to analyze the applications for compatibility with the intended device. Debugging programs have been used in the past to provide some analysis capabilities. However, current debugging programs do not provide flexibility in selecting platforms and applications to analyze, in sorting and displaying the results, and

25   in providing graphical representations of comparison results.

SUMMARY OF THE INVENTION

The present invention comprises a method and apparatus for receiving data analysis instructions from a user and for displaying results of the data

5   analysis to the user. In one embodiment, the invention provides a series of interface windows displayed on a computer display screen. A first window allows a user to select one or more specification files comprising reference data specifying capabilities and or resources with which other data is to be compared. A second window allows the user to select one or more data files to be compared

10   to the one or more specification files selected using the first window. A third window allows the user to interactively select the type of analysis to be performed and provides the results of the selected analysis to the user. The third window also allows the user to interactively apply one or more filters to the results and to select the manner (view) in which the results are displayed

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows an example of a display screen used in one or more embodiments of the invention.

5

Figure 2 shows an example of a display screen used in one or more embodiments of the invention.

Figure 3 shows an example of a display screen used in one or more

10   embodiments of the invention.

Figure 4 shows an example of a display screen used in one or more embodiments of the invention.

15   Figure 5 shows how results may be reordered in one or more embodiments of the invention.

Figure 6 shows how results may be reordered in one or more embodiments of the invention.

20

Figure 7 shows an example of a display screen used in one or more embodiments of the invention.

Figure 8 is a flow chart for one or more embodiments of the invention.

25

Figure 9 is a block diagram of an example computer system that may be used to implement one or more embodiments of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus for analyzing data is described. In the following description, numerous specific details are set forth in order to provide a more thorough description of the invention. It will be apparent, however, to one skilled in the art that the invention may be practiced without these specific details. In other instances, well-known features have not been described in detail so as not to obscure the invention.

Figure 1 shows an example of a display screen used in one or more embodiments of the invention. The display screen of Figure 1 is used in an embodiment of the invention that compares data comprising computer program code written in the Java programming language with data describing the specifications of different types of Java operating environments. For example, different implementations of Java exist for different types of hardware platforms, ranging from "smartcards" to supercomputers. Examples of scaled down implementations of Java include Personal Java for use in consumer devices for home, office and mobile use and Embedded Java for use in embedded systems. Personal Java and Embedded Java do not provide the same range of resources as more complete forms of Java, such as, for example, Business Java. It is useful for a developer to know whether a Java program is compatible with (i.e. can be executed using) the different Java platforms. The embodiment of Figure 1 compares user selected program files with user selected operating environment specifications and informs the user of the degree of compatibility between the two. The operation of the program is described in copending U.S. Patent Application Serial No. _____ for Method and Apparatus for Assessing

Compatibility Between Platforms and Applications filed on September 29, 1997, and incorporated by reference herein.

The display screen 100 of Figure 1 may comprise the whole screen of a
5   computer display or may constitute a resizable window. As shown in Figure 1, display screen 100 includes a title bar 105, a status message area 110, page tabs 115a-d, and page display area 120.

Title bar 105 displays a title, for example the name of the program that
10   displays display screen 100. Status message area 110 displays messages indicating status, problems, instructions or other information. Page tabs 115a, 115b, 115c, and 115d are used to select a corresponding page for display in page display area 120. Page tab 115a selects a page that is used by the user to select the file or files containing the specifications of the Java implementations for which
15   the user wants to determine the compatibility of Java programs. Page tab 115b selects a page that is used by the user to select the file or files containing the program code that the user wants to be tested for compatibility with the Java implementations selected by the user using page tab 115a. Page tab 115c selects a page that is used by the user to select the type of analysis to be performed and
20   that displays the results of the analysis to the user. Page tab 115d selects a "Help" page.

In Figure 1, page tab 115a is shown as having been selected. Accordingly, a specification file input page 125 is displayed in page display area 120. As
25   shown in Figure 1, specification file input page 125 includes a directory display area 130, a specification file display area 135, and a message display area 140.

Directory display area 130 displays file directory information. It is used by the user to select directories in which specification files are located, and then to select specification files from the displayed directory. For example, in the

5   example of Figure 1, the user has selected the directory "/home/looney/projects/specs", as shown in directory path information area 145. The files in the selected directory, as shown in directory display area 130, are named "cjava," "ejava," and "pjava." In this example, these files contain specifications for different Java platforms. In other embodiments, the

10   specification files may contain specifications for other types of resources or requirements. A user may select one or more of specification files shown in directory display area 130 by any of a variety of well known selection techniques. For example, the user may select a file name by clicking on the name using a mouse or other pointing device and then clicking "add" button 150, or the user

15   may simply double-click on a file name. Any of a variety of other selection techniques may be used. After a specification file is selected by the user, it is listed in specification file display area 135.

Message display area 140 is used to display informational messages to the

20   user regarding the specification files the user has selected. The messages may include error, status, warning and other informational messages. For example, a warning message may be displayed if the file that was selected is not a proper specification file or if a specification file has been corrupted. Information concerning the content of selected specification files may also be displayed. For

25   example, if more than one specification file has been selected, a message may be

displayed that indicates differences between the specification files (e.g. differences in classes supported).

Figure 2 shows specification input page 125 of Figure 1 after two of the

5   specification files listed in directory display area 130 have been selected by the user. In the example of Figure 2, the selected specification files are listed in order of selection by the user, and directory path information for each file is displayed. In the example of Figure 2, a specification file can be removed from specification file display area 135 by selecting the file using a pointing device and clicking on

10   "remove" button 200.

Figure 3 shows a program file selection page 300 that is displayed when page tab 115b of the example of Figure 1 is selected. As shown in Figure 3, file selection page 300 includes a directory display area 330 and a program file

15   display area 335.

Directory display area 330 of file selection page 300 is used in the same manner as directory display area 130 of specification file input page 125 to select program files that the user wishes to test for compatibility with the specification

20   files selected by the user using specification file input page 125. The user may select the program files using file selection page 300 before or after selecting specification files using specification input page 125. The files selected by the user for testing are listed in program file display area 335. A file may be removed from program file display area 335 by selecting the file using a pointing

25   device and clicking "remove" button 340.

Figure 4 shows a data analysis page 400 that is displayed when page tab 115c of Figure 1 is selected. In order to use data analysis page 400, at least one specification file and one program file must first have been selected by the user using specification file input page 125 and file selection page 300. Otherwise, an

5   error message will be displayed in status message area 110.

In the example of Figure 4, data analysis page 400 includes an analysis criteria selection area 410, a result display area 415, and an information display area 420.

10   Analysis criteria selection area 410 contains interface controls that allow a user to specify the type of information that the user wishes to obtain. In the example of Figure 4, analysis criteria selection area 410 includes four check boxes 411, 412, 413 and 414, respectively, that allow the user to select the kind of error information that the user wishes to have displayed in result display area

15   415. A user may select a check box, for example, by clicking on it using a pointing device such as a mouse. In the example of Figure 4, the analysis criteria comprises various filters that can be applied to the analysis. In other embodiments, other types of analysis criteria may be specified. In Figure 4, check box 411 allows the user to display "unsupported errors." Check box 412

20   allows the user to display "modified errors." Check box 413 allows the user to display "optional errors." Check box 414 allows the user to display "unresolved errors."

In one or more embodiments of the invention, errors are identified as follows. An attempt is made to resolve application references within the

25   application. However, the application may contain external references that

cannot be resolved within the application. If neither the application nor the operating platform contains an element that can resolve a reference, the reference is unresolvable.

5   If a reference cannot be resolved within the application, the merged specification is examined to determine whether the operating platform contains a programming resource that can be used to resolve the reference. If the programming resource is found, a compliance status associated with the programming resource is examined to determine whether it is available for use

10   by the application. If the programming resource has a compliance status of "required," the operating platform is required to include the programming resource and the application's reference is reported as resolved. An "optional" compliance status indicates that the entry may or may not be included in the operating platform. If the optional status is not overridden by a

15   subsequently-processed specification, the reference is reported in error (e.g., the reference is to an unimplemented optional programming resource). A "modified" compliance status indicates that the programming resource associated with a the entry has been changed. An "unsupported" compliance status indicates that the application's reference is not supported by the operating

20   platform. A status of "optional," "modified" or "unsupported" resolves the reference, but indicates a potential problem with that reference.

Either a warning or error message is generated for those references that are not resolved to the application programming resource or a "required"

25   specification programming resource. A "non-conformance" warning is generated for those references that resolve to a "modified" specification

programming resource, or for a reference that does not resolve to a programming resource (i.e., an "unresolved" status). A reference that resolves to an "unsupported" or "optional" specification programming resource causes a "non-conformance" error.

5

In the example of Figure 4, all four check boxes are shown as having been selected, which is the default display mode for the embodiment of Figure 4. Accordingly, all four types of errors will be displayed in result display area 415, unless the user elects to uncheck one or more boxes.

10

The user may initiate the data analysis by clicking "analyze classes" button 440 ("class" is a name used for Java program files). The results of the analysis (i.e. testing the user selected program files for compatibility with the user selected specification files) are displayed in result display area 415. In addition, in the

15 example of Figure 4, the number of each of the error types selected by the user that are found during the analysis is displayed next to the error type name in analysis criteria selection area 410, and a message indicating information about the analysis performed (e.g. "The class files analyzed do not conform to the specification files") is displayed in status message area 110.

20

In the example of Figure 4, the results are displayed in chart form in result display window 415. In Figure 4, the chart includes three columns 445, 450, and 455, respectively. Each column includes a title bar that identifies the content of the column. Title bar 425 identifies the content of column 445 as "classes

25 analyzed." Title bar 430 identifies the contend of column 450 as "errors." Title bar 435 identifies the content of column 455 as "dependencies." Each row of the

chart corresponds to a particular result: either an error indication or an "OK" indication. Column 445 lists the program (class) file name to which the result displayed in a row pertains. Column 450 identifies the result (by specifying either "OK" or the type of error). Column 455 identifies the particular element of

5   the program file that is the cause of an error specified in column 450 (e.g. a resource or dependency that is required by the program file but that is lacking in the specification file to which the program file is being compared).

Data analysis page 400 allows the user to interactively filter and sort the

10   data results that are displayed in result display area 415. For example, the user may change the kinds of errors that are displayed by checking or unchecking one or more of check boxes 411-414 in analysis criteria selection area 410. The user may also change the order in which the results are displayed in result display area 415 by clicking on any of title bars 425, 430, and 435. Clicking on a title bar

15   sorts the results listed in result display area 415 by the contents of the column to which the title bar corresponds. For example, in Figure 4, title bar 425 of column 445 is shown as having been selected by the user. Accordingly, the results shown in result display area 415 are sorted by program (class) file name. Figure 5 shows how the results of Figure 4 are reordered when title bar 430 is selected such that

20   the results shown in result display area 415 are sorted according to the error type listed in column 450. Figure 6 shows how the results of Figure 4 are reordered when title bar 435 is selected such that the results shown in result display area 415 are sorted according to the error identification information listed in column 455.

25

Referring again to Figure 4, information display area 420 is used to display additional result information when a particular result listed in result display area 415 is selected. For example, if the user clicks on entry 470 in column 455, additional explanatory information, if any, available for the entry

5   "java.io.Zajajjfa" is displayed in information display area 420.

A user may save the results displayed on data analysis page 400 in a file by clicking "save" button 475.

10   Figure 7 shows an example of a help page 700 that is displayed in page display area 120 when page tab 115b of Figure 1 is selected by the user. Help page 700 displays help information as is well known in the art.

Figure 8 is a flow chart for a process used in one or more embodiments of

15   the invention. In the process of Figure 8, the program begins at block 800. At block 805, the initial screen of the program is displayed. The initial screen may, for example, be screen 100 of Figure 1, and preferably includes page tabs such as tabs 115a-d and a page display area such as page display area 125 that displays different user interface pages corresponding to the page tab that has been

20   selected. In the example of Figure 8, the initial page displayed in the page display area of the screen when the program is first activated is a specification input page, for example specification input page 125. A default start-up status message is also displayed in a status message display area, such as message display area 110.

25

At block 810, user input is awaited. After user input is received, a determination is made at block 815 as to whether a different page tab from the currently selected page tab has been selected. If not, additional user input is awaited at clock 810.

5

If a different page tab has been selected, a determination is made at block 820 as to whether the page tab that has been selected is a data analysis page tab, such as data analysis page tab 115c. If it is determined that a data analysis page tab has been selected, the data analysis page is displayed in the page display area

10 at block 825.

At block 830, a determination is made as to whether at least one program file and one specification file have already been selected by the user. If yes, additional user input is awaited at block 835. If not, an error message is

15 displayed in the status message display area at block 845, and then user input is awaited at block 835.

After user input is received at block 835, a determination is made at block 840 as to whether a different page tab has been selected. If yes, processing

20 continues to block 820. If not, a determination is made at block 850 as to whether an analyze button shown on the data analysis page, such as analyze button 440, has been activated. If not, processing returns to block 835. If yes, data analysis corresponding to criteria indicated by the user's input is performed and the results displayed to the user at block 855. From block 855, processing continues

25 to block 835.

If at block 820 it is determined that the new page tab selected is not the data analysis page tab, a determination is made at block 865 as to whether the page tab that has been selected is a program file selection page tab, such as program file selection page tab 115b. If yes, a program file selection page, such as

5   program file selection page 300, is displayed in the page display area at block 860, and processing continues to block 810. If not, processing continues to block 875.

At block 875 a determination is made as to whether the page tab that has

10   been selected is a specification file input page tab, such as specification file input page tab 115a. If yes, a specification file input page, such as specification file input page 125, is displayed in the page display area at block 870, and processing continues to block 810. If not, processing continues to block 885.

15   At block 885 a determination is made as to whether the page tab that has been selected is a help page tab, such as help page tab 115d. If yes, a help page, such as help page 700, is displayed in the page display area at block 880, and processing continues to block 810. If not, processing continues directly to block 810.

20

The invention may be implemented by means of appropriate software on the computer system of Figure 9. The exemplary computer system shown in Figure 9 includes a CPU unit 900 that includes a central processor, main memory 910, peripheral interfaces, input-output devices, power supply, and associated

25   circuitry and devices; a display device 930 which may be a cathode ray tube display, LCD display, gas-plasma display, or any other computer display; an

input device 950, which may include a keyboard, mouse, digitizer, or other input device; non-volatile storage 920, which may include a floppy disk drive, a RAM card, hard drive, CD-ROM drive, or other magnetic, re-writable optical, or other mass storage devices; and a network interface 940, which may include a modem,

5   allowing the computer system to communicate with other systems over a communications network such as the Internet. Any of a variety of other configurations of computer systems may also be used.

Thus, a method and apparatus for analyzing data has been presented.

10   Although the present invention has been described with respect to certain example embodiments, it will be apparent to those skilled in the art that the present invention is not limited to these specific embodiments. Further, although the operation of certain embodiments has been described in detail using certain detailed process steps, some of the steps may be omitted or other

15   similar steps may be substituted without departing from the scope of the invention. Other embodiments incorporating the inventive features of the present invention will be apparent to those skilled in the art. The invention is defined by the claims and their full scope of equivalents.

20